

## **REMARKS**

Applicant is in receipt of the Office Action mailed January 6, 2005. Reconsideration of the present case is earnestly requested in light of the following remarks.

### **§103 Rejections**

Claims 1-7 and 9-36 were rejected under 35 U.S.C. 103(a) as being unpatentable over Leask et al. (U.S. Patent No. 6,412,106, hereinafter "Leask") in view of Kodosky (U.S. Patent No. 5,481,740). This rejection is respectfully traversed.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest the features as recited in claim 1:

A computer-implemented method for creating a graphical program,  
the method comprising:

creating a first graphical program, wherein said creating comprises interconnecting at least two of a first plurality of graphical program nodes or icons, wherein the first graphical program comprises the first plurality of interconnected graphical program nodes or icons which graphically represents functionality of the first graphical program, and wherein the first graphical program is executable by a computer system to perform the functionality;

storing the first graphical program in a memory; and

associating a debugging graphical program at a debugging location in the first graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program, and wherein the debugging graphical program is executable by the computer system to perform the functionality;

wherein the debugging graphical program is executable during execution of the first graphical program to aid in debugging at least a portion of the first graphical program.

The Examiner asserts “However, Kodosky in an analogous art teaches a panel palette in a graphical debugging environment which includes command buttons or icons that graphically represent the functionality of the debugging program (35:15-25). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Kodosky’s limitations of providing interconnected graphical icons that graphically represent functionality of the debugging with Leask. . .” (*emphasis added*). Applicant respectfully submits that these “command buttons or icons” in a palette are not interconnected graphical program nodes or icons that graphically represent functionality of the debugging graphical program.

Rather, Kodosky teaches:

#### Panel Palette

Both the Panel and Diagram windows contain a palette of command buttons and status indicators that are used for controlling the VI. One of two palettes is available depending on whether the user is operating in run or edit mode. The palette illustrated in FIG. 48 appears at the top of the window when the VI is in run mode. This panel includes, from left to right, icons referred to as the Run button, the Mode button, the Continuous Run button, the Breakpoint button, the Step Mode button, the Execution Highlighting button, the Print Mode button, and the Datalogging button. (Kodosky col. 35, lines 15-25) (*emphasis added*)

Furthermore, Kodosky’s FIG. 48 illustrates the “palette of command buttons” which shows that the “command buttons” are neither connected nor interconnected to each other.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest “. . . associating a debugging graphical program at a debugging location in the first

graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program, and wherein the debugging graphical program is executable by the computer system to perform the functionality. . .” as recited in claim 1.

Applicant respectfully notes “To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974)” MPEP §2143.03 (*emphasis added*).

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 1. Accordingly, Applicant respectfully submits that, at least for one or more reasons presented, claim 1 and those dependent therefrom are allowable.

Furthermore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest “. . . wherein said associating the debugging graphical program at the location in the first graphical program comprises associating the debugging graphical program at a first data flow path in the first graphical program” as recited in claim 13.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 13.

The Examiner asserts “Regarding claim 13, the computer-implemented method of claim 12 [sic], wherein the first graphical program comprises a plurality of data flow paths (Leask, 12:18-20, call flow)” (*emphasis added*). Applicant respectfully disagrees with the Examiner’s characterization of Leask and respectfully submits that a “call flow”, as referenced in Leask, is a program flow. Leask in col. 12, lines 18-20 references “call flow diagram 300” to Leask’s FIG. 4. Moreover, Leask teaches:

FIGS. 3 and 4 illustrate examples of a graphical development environment. FIGS. 3 and 4 illustrate exemplary screens that may be presented to a developer when designing a call flow. An example of such a graphical development environment is provided in co-pending, commonly

assigned U.S. application Ser. No. 08/599,134 filed Feb. 9, 1996, entitled "ENHANCED GRAPHICAL DEVELOPMENT ENVIRONMENT FOR CONTROLLING PROGRAM FLOW." (Leask col. 10, lines 50-58)

U.S. Application Serial No. 08/599,134 filed Feb. 9, 1996 became U.S. Patent No. 5,946,485 to Weeren et al. (hereinafter, "Weeren"). Applicant submits the Weeren reference in an Information Disclosure Statement filed herewith.

Applicant respectfully submits that Weeren teaches and discloses that a call flow is a program flow: "Such graphical development environments used icons to represent various language components. Developers would draw lines or arrows connecting these icons. These lines defined a program flow (call flow)" (Weeren col. 1, lines 44-46) (*emphasis added*). Weeren further teaches that control of a program from start to finish is also a "call flow": "A preferred embodiment of the graphical development environment of the present invention is used to develop programs for voice response systems in the telecommunications environment. Accordingly, this discussion refers to the flow of the program from start to finish as the 'call flow'" (Weeren col. 4, lines 58-62) (*emphasis added*).

Accordingly, Applicant respectfully submits that Leask nowhere teaches or suggests data flow. Therefore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest "...wherein said associating the debugging graphical program at the location in the first graphical program comprises associating the debugging graphical program at a first data flow path in the first graphical program" as recited in claim 13.

Thus, Applicant respectfully submits that claim 13 is allowable for these further reasons.

Furthermore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest: "The computer-implemented method of claim 13, wherein said associating comprises: storing information in at least one data structure, wherein the information comprises information regarding the first graphical program, the debugging graphical program, and the location where the debugging graphical program is attached along the

first data flow path of the first graphical program (*emphasis added*)” as recited in claim 14.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 14.

Applicant respectfully submits that the arguments presented above which submit that Leask nowhere teaches or suggests data flow apply with equal force to claim 14, as well. Accordingly, Applicant respectfully submits that claim 14 is allowable for these further reasons.

Furthermore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest the features of claim 15:

The computer-implemented method of claim 13, wherein said associating comprises:

receiving user input from a pointing device selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type;

displaying a plurality of debugging graphical programs, wherein each of the plurality of debugging graphical programs is compatible with the first data type, and wherein the plurality debugging graphical programs comprises the debugging graphical program; and

receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs.

Applicant respectfully submits that the arguments presented above which submit that Leask nowhere teaches or suggests data flow apply with equal force to claim 15, as well.

Moreover, the Examiner asserts “Regarding claim 15. . .receiving user input from a pointing device selecting a data flow path in the first graphical program wherein the first data flow path is configured to carry data of a first data type;. . .” without pointing to any evidence in the record supporting this finding. Applicant respectfully notes MPEP 707.07(d) which requires that, in an Examiner’s Action, the ground of rejection, should

be “fully and clearly stated”. Moreover, “[T]he the Board [or the Examiner] must point to some concrete evidence in the record in support of these findings.” *In re Zurko*, 258 F.3d 1379, 1386 (Fed. Cir. 2001) (*emphasis added*).

The Examiner further asserts “display a plurality of graphical programs (Leask, 21:40-43, and 55-60, for multiple debug tools); and . . .” Applicant is uncertain if the Examiner considered the entire feature “. . .displaying a plurality of debugging graphical programs, wherein each of the plurality of debugging graphical programs is compatible with the first data type, and wherein the plurality debugging graphical programs comprises the debugging graphical program. . .” as recited in claim 15. Furthermore, the term “data type” appears nowhere in Leask.

Further, the Examiner asserts: “receiving user input selecting one of the debugging graphical programs (Leask, 21:47-50, see touch tone for user input)” (*emphasis added*). Leask teaches “touch tones” may be user input received by an application while it is being debugged:

Debugging an application program during runtime offers developers the ability to analyze the program's operation to detect and examine problems with the program without interrupting the program's operation. For example, suppose that a VRU application is running at a bank which allows the bank's customers to call the VRU and interact with the application by pressing touch tone keys on the customers' telephones or speaking verbal commands into the telephones. (Leask col. 21, lines 43-51) (*emphasis added*)

While Leask teaches “touch tones” may be user input received by an application while it is being debugged, Leask nowhere teaches or suggests “. . .receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs” as recited by claim 15.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest the features of claim 15. Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 15. Accordingly, Applicant respectfully submits that, at least for these further reasons, claim 15 is allowable.

Furthermore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest the features of claim 16:

The computer-implemented method of claim 13, wherein said associating comprises:

receiving user input selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type;

determining the first data type of the first data flow path;

displaying a plurality of debugging graphical programs appropriate for the first data type of the first data flow path; and

receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data type of the first data flow path.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest the features of claim 16. Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 16.

Applicant respectfully submits that the arguments presented above which submit that Leask nowhere teaches or suggests data flow apply with equal force to claim 16, as well.

Further, the Examiner asserts “Regarding claim 16. . .receiving user input selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type (Leask, FIG. 8, 810, and associated text);. . .” (*emphasis added*).

Instead of receiving user input in Leask’s FIG. 8, Leask teaches teaches interaction between two sets of program instructions rather than receiving user input: “Turning now to FIG. 8, an exemplary flow diagram illustrating the interaction between the debug engine and the graphical debugger in conducting debug tool management (e.g., inserting a debug tool, removing a debug tool, and changing attributes of a debug tool) is

provided” (Leask col. 20, lines 9-13) (*emphasis added*). Furthermore, Applicant respectfully submits that the term “data type” appears nowhere in Leask.

In contrast, Applicant’s invention as recited in claim 16 includes “. . .receiving user input selecting the first data flow path in the first graphical program, wherein the first data flow path is configured to carry data of a first data type. . .”. Leask and Kodosky, taken both singly and in combination, nowhere teach or suggest this feature.

Moreover, the Examiner asserts “Regarding claim 16. . .determining the first data type of the first data flow path;. . .” without pointing to any evidence in the record supporting this finding. Applicant respectfully notes MPEP 707.07(d) which requires that, in an Examiner’s Action, the ground of rejection, should be “fully and clearly stated”. Moreover, “[T]he Board [or the Examiner] must point to some concrete evidence in the record in support of these findings.” *In re Zurko*, 258 F.3d 1379, 1386 (Fed. Cir. 2001) (*emphasis added*).

The Examiner also asserts: “. . .displaying a plurality of debugging graphical programs appropriate for the first data type of the first data flow path (Leask, 21:40-43, and 55-60, for multiple debug tools);. . .” (*emphasis added*). Rather, in col. 21, lines 54-61, Leask teaches that an application may be debugged while it is still running a production environment without interrupting operation of the application.

Moreover, Applicant respectfully submits that Leask does not teach or suggest a plurality of debugging graphical programs appropriate for a first data type of a first data flow path, rather Leask teaches a single “graphical debugging environment” with an ability to include multiple debugging tools into an application:

It should be understood that the graphical debugging environment allows debug tools to be set for multiple block IDs within an application program. That is, the flow diagrams of FIGS. 7-9 are not limited to merely 1 debug tool being requested for an application program, but multiple debug tools may be requested within an application program. Accordingly, each debug tool requested by a developer may have an indicator displayed within the graphical representation of the application program and may be communicated to the debug engine. (Leask col. 21, lines 33-42) (*emphasis added*)



In contrast, Applicant's invention as recited in claim 16 includes “. . .displaying a plurality of debugging graphical programs appropriate for the first data type of the first data flow path. . .” Neither Leask nor Kodosky teach or suggest this feature.

The Examiner further asserts “. . .receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data flow path (Leask, 26:59-65, also see Kodosky, 18:27-35).”

In col. 26, lines 60-65 Leask teaches receiving user input to interact with a graphical representation of a computer application: “said program code comprises code responsive to user input for interacting directly with said graphical representation of said computer application program to debug said computer application program, wherein interaction with the textual source code of said computer application program is not required.” However, Leask is deficient in teaching or suggesting “. . .receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data type of the first data flow path (*emphasis added*)” as recited in claim 16.

Furthermore, Kodosky teaches a “Case Structure Node” in col. 18, line 18 – col. 19, line 5 which executes various portions of program instructions if a respective condition is satisfied. Even if the Examiner interprets selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data type of the first data flow path as a possibility of a “Case Structure Node”, Applicant respectfully notes “Inherency, however, may not be established by probabilities or possibilities.” *In re Robertson*, 169, F.3d 743, 745 (Fed. Cir. 1999) (quoting *Continental Can Co. v. Monsanto Co.*, 948 F.2d 1264, 1268, 20 U.S.P.Q.2d 1746, 1749 (Fed.Cir.1991) (quoting *In re Oelrich*, 666 F.2d 578, 581, 212 U.S.P.Q. 323, 326 (C.C.P.A.1981))) (*emphasis added*).

Therefore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest, either expressly or inherently, “. . .receiving user input selecting the debugging graphical program from the plurality of debugging graphical programs appropriate for the first data type of the first data flow path” as recited by claim 16.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 16. Accordingly, Applicant respectfully submits that, at least for these further reasons, claim 16 is allowable.

Furthermore, Applicant respectfully submits that neither Leask nor Kodosky teach or suggest Applicant's invention as recited in claim 37:

The computer-implemented method of claim 1, wherein said associating the debugging graphical program at the debugging location in the first graphical program comprises:

including the debugging graphical program at the debugging location in the first graphical program.

Rather, Leask teaches “. . .the graphical debugging environment allows a user to insert debugging tools such as breakpoints, directly into the graphical representation of the application program” (Leask Abstract) (*emphasis added*). As Applicant submits above, “the graphical representation of the application program” is not a first graphical program comprises the first plurality of interconnected graphical program nodes or icons which graphically represents functionality of the first graphical program, wherein the first graphical program is executable by a computer system to perform the functionality. Furthermore, neither “the graphical debugging environment” nor “debugging tools such as breakpoints” are a debugging graphical program comprises a second plurality of connected graphical program nodes or icons that graphically represents functionality of the debugging graphical program.

Moreover, even if Leask does teach inserting a plurality of breakpoints into a graphical representation of an application, the plurality of breakpoints are not interconnected with each other, thereby creating a plurality of interconnected breakpoints.

Thus, Applicant respectfully submits that claim 37 is patentably distinguished over both Leask and Kodosky, taken both singly and in combination. Accordingly, Applicant respectfully submits that, at least for these further reasons, claim 37 is allowable.

Applicant respectfully submits that each of claims 38-40 include features similar to claim 37, and so the arguments presented above apply with equal force to each of these claims, as well. Thus, Applicant respectfully submits that each of claims 38-40 is patentably distinguished over both Leask and Kodosky, taken both singly and in combination. Accordingly, Applicant respectfully submits that, at least for these further reasons, each of claims 38-40 is allowable.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest “. . .providing the data to a debugging graphical program, wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program, and wherein the debugging graphical program is executable by the computer system to perform the functionality. . .”, “. . .executing the debugging graphical program, wherein the debugging graphical program uses the data. . .”, and “. . .the debugging graphical program generating debugging results” as recited in claim 23.

Applicant respectfully submits Leask teaches controlling a program flow of an application (Leask col. 10, lines 50-58) rather than providing data to a debugging graphical program. Furthermore, neither Leask nor Kodosky teach or suggest “. . .wherein the debugging graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the debugging graphical program. . .” as recited in claim 23.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 23. Accordingly, Applicant respectfully submits that, at least for the reasons presented, claim 23 and those dependent therefrom are allowable.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest “. . .associating a second graphical program at a location in the first graphical program, wherein said associating does not modify the functionality of the first graphical program, wherein the second graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the second graphical program, and wherein the second graphical program is executable by the

computer system to perform the functionality. . .” and “. . .wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program” as recited in claim 27.

Furthermore, Applicant respectfully submits one or more arguments presented above for claim 1 are applicable to claim 27, as well.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 27. Accordingly, Applicant respectfully submits that, at least for the reasons presented, claim 27 and those dependent therefrom are allowable.

Claim 31 includes limitations similar to claim 27, and so the arguments presented above apply with equal force to claim 31, as well. Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 31. Accordingly, Applicant respectfully submits that, at least for the reasons presented, claim 31 and those dependent therefrom are allowable.

Applicant respectfully submits that neither Leask nor Kodosky teach or suggest “. . .a second graphical program, wherein the second graphical program comprises a second plurality of interconnected graphical program nodes or icons that graphically represents functionality of the second graphical program, and wherein the second graphical program is executable by the computer system to perform the functionality. . .”, “. . .a data structure which is operable to store information associating the second graphical program with a location in the first graphical program, wherein the functionality of the first graphical program is not modified by the second graphical program. . .”, and “. . .wherein the second graphical program is executable during execution of the first graphical program to aid in analyzing at least a portion of the first graphical program” as recited by claim 35.

Furthermore, Applicant respectfully submits that one or more arguments presented above for claim 1 are applicable to claim 35, as well.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 35. Accordingly, Applicant respectfully submits that, at least for the reasons presented, claim 35 and those dependent therefrom are allowable.

Claim 8 was rejected under 35 U.S.C. 103(a) as being unpatentable over Leask in view of Kodosky as applied to claim 7 in view of McKee et al. (U.S. Patent No. 5,915,114, hereinafter “McKee”). This rejection is respectfully traversed.

Independent claim 1 has been argued to overcome rejections under 35 U.S.C. 103. Above, Applicant respectfully submits that claim 1 is nonobvious and allowable. Applicant respectfully notes: “If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)” as stated in the MPEP §2143.03. Accordingly, Applicant respectfully submits that, at least for that reason, claim 8 is allowable.

Furthermore, Leask, Kodosky, and McKee, taken both singly and in combination, nowhere teach or suggest “. . .data generated by the debugging graphical program. . .” as recited by claim 8. Rather, McKee a “dynamic trace-driven code optimizer” (McKee Abstract).

Moreover, McKee nowhere teaches or suggests a graphical program which comprises a plurality of interconnected graphical program nodes or icons which graphically represents functionality of the graphical program, and McKee does not teach or suggest “. . .wherein said differences in execution times are useable in optimizing performance the first graphical program (*emphasis added*)” as recited by claim 8.

Thus, Applicant respectfully submits that a *prima facie* case of obviousness has not been established to reject claim 8. Accordingly, Applicant respectfully submits that, at least for these further reasons presented, claim 8 is allowable.

Applicant also respectfully submits that there is no teaching, suggestion, or motivation to combine Leask, Kodosky, and/or McKee either in the references or in the prior art. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolchem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to

produce the claimed invention is defective as hindsight analysis. Furthermore, Applicant respectfully submits that it is nonobvious to combine Leask, Kodosky, and/or McKee.

Moreover, the showing of a suggestion, teaching, or motivation to combine prior teachings “must be clear and particular. . .Broad conclusory statements regarding the teaching of multiple references, standing alone, are not ‘evidence’.” *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination. Applicant respectfully submits that there is no suggestion in the prior art for combining Leask, Kodosky, and/or McKee, and that even if the references were combined, they would not produce the features of claims 1-11 and 13-40.

Applicant respectfully requests removal of the §103 rejection.

## CONCLUSION

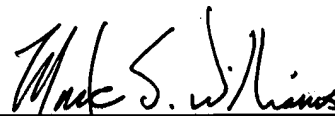
In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-59901/JCH.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☒ Request for Continued Examination
- ☒ Information Disclosure Statement

Respectfully submitted,



---

Mark S. Williams  
Reg. No. 50,658  
Agent for Applicant

Meyertons, Hood, Kivlin, Kowert & Goetzel PC  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8800  
Date: April 6, 2005 JCH/IMF